

ICICIMICICICONCENT

JLINK调试仿真及烧写FLASH教程



This is j-link



JTAG: 国际标准测试协议 RDI: ARM公司提出的调试接口标准

为什么要使用JLINK?

- 1、JLINK用硬件进行协议转换,烧写、仿真速度快。
- 2、支持的芯片多。
- 3、JLINK使用USB下载线与计算机相连,仿真、烧 写程序非常方便。

Part 1 JLINK+AXD+S3C4510B调试仿真 A、配置ADS工程

B、编译并进入AXD

X

C、配置AXD

D、调试仿真



高级语言程序从<mark>源代码</mark>到成为可在硬件上运行的可执行代码需要经历 四个阶段:



因此,我们配置ADS工程主要配置编译器和链接器。



主要配置的选项:

ARM Assembler (编译器)
 ARM C Compiler (编译器)
 ARM Linker (链接器)

其他选项默认即可。



1、ARM Assembler

| DebugRel Settings | ? × |
|--|---|
| Target Settings Panels | ARM Assembler |
| Target Target Settings Access Paths Build Extras Runtime Settings File Mappings | Target ATPCS Options Predefines Listing Control Extras Architecture or Processor Eloating Point ARMTIDMI Pure-endian softfp |
| - Source Trees - ARM Target - Language Settings - ARM Assembler - ARM C Compiler - ARM C Compiler | Byte Order |

目的:选择与ARM核相匹配的汇编代码编译器



2、ARM C Compiler



目的:选择与ARM核相匹配的C代码编译器



3、ARM Linker (Output)

| Target Settings Panels | ARM Linker | | |
|---|---|---|--|
| - Language Settings 🔺 | Output Options L | ayout Listings Extras | |
| ARM C Compiler ARM C++ Compiler Thumb C Compiler Thumb C++ Com | Linktype <u>P</u> artis Simp <u>l</u> e <u>S</u> attered | Simple image RO Base RW Base Rv Ox400000 Rv Sp | opi 🦳 Reloc <u>a</u> tabl /P <u>i</u> olit I <u>m</u> ag |
| Linker ARM Linker ARM fromELF | <u>S</u> catter Symbol | | Choose |
| - Editor - Cu <u>st</u> om Keywords | Symbol edition | | Choose |

目的: 设置代码段的起始地址为RAM的起始地址。

在ARM的集成开发环境中,只读的代码段和常量被称作R0段(ReadOnly); 可读写的全局变量和静态变量被称作RW段(ReadWrite);RW段中要被初始化为 零的变量被称为ZI段(ZeroInit)。



3、ARM Linker (Layout)

| Target Settings Panels | ARM Linker | - |
|--|--------------------------------|---------|
| - Language Settings 🔺 - ARM Assembler - ARM C Compiler | Output Options Layout Listings | Extras |
| ARM C++ Compiler | Object/Symbol | Section |
| - Thumb C Compiler Thumb C++ Com | init.o | init |
| ARM Linker | -Place at end of image | |
| ARM fromELF | O <u>b</u> ject/Symbol | Section |
| - Custom Keywords | | |

目的: 1、指定放置在可执行文件开头的目标文件为init.o 2、指定放置的逻辑段的段名为init(本例中init为代码段)



init.o?

汇编结束后,生成多个目标文件,一般是一个源代码文件生成一个 目标文件(头文件除外),然后由链接器来把这些目标文件链接成一个 可执行的二进制代码文件。这个文件可用来调试或者烧写到ROM中。





配置完毕之后一定要注意存盘(ctrl+s), 这样配置才会生效,否则编译时会报错。

编译并进入AXD

菜单>Projiect>Debug

| <u>E</u> dit <u>V</u> iew <u>S</u> earch | Project | Debug | Window | <u>H</u> elp | 157 |
|--|----------------|--------------------|--------------|---------------|-----|
| - 🖻 🔳 🗠 🤄 | <u>A</u> dd n | ain.c to | Project | | |
| | Add E | iles | | | |
| nain.c | Create | e Group | | | |
| | Create | e Target | 110 | | |
| • {} • M. • 🖻 • | Create | e Segme | nt/Overlay | | |
| define IOPMOD | Check | : Synta <u>x</u> | | Ctrl+; | F |
| define IOPCON | Prepro | ocess | | | F |
| define IOPDAT | Pr <u>e</u> co | mpile | | | F |
| oid delay(un | <u>C</u> ompi | ile | | Ctrl+F7 | |
| int main(woid) | Djsas | semble | | Ctrl+Shift+F7 | |
| (| Bring | Up To <u>D</u> a | ate | Ctrl+U | |
| IOPMOD=0X, while(1) | Make | | | F7 | |
| { | Stop B | Build | | Ctrl+Break | |
| TOPDATA=UX delav(500) | Remo | ve Objec | t Code | Ctrl+- | |
| IOPDATA=0X | Re-se | arch for | Files | | |
| delay(500) 3 | Reset | Project | Entry Paths | 3 | |
| return 0; | Synch | nroni <u>z</u> e N | Aodification | n Dates | |
| t. | Debug | 1 | | F5 | |
| | Run | | | Ctrl+F5 | |
| void delay(uns | Set De | efault Pro | oiect | | |
| ۱ unsigned | Set De | efault Ta | raet | | E. |

命令介绍 Compile: 编译单个源文件,生成一个.o文件。 Make: 编译整个工程,生成多个.o文件和 一个.axf文件。 **Debug:** 编译整个工程,生成多个.o文件和 一个.axf文件,并进入AXD。 Run: 编译整个工程,生成多个.o文件和 一个.axf文件,并进入AXD,同时 ;运行程序。



AXD的配置步骤只需以下两步:

一、加载初始化指令

二、加载JLINK动态链接库





一、加载初始化指令

新建一个文本文档,输入下图所示的初始化指令,然后保存以备用。
 菜单>Options>Configure Interface,加载刚才的初始化文本。





二、加载JLINK动态链接库

菜单>Options> Target,加载动态链接库JLinkRDI.dll,该文件在J-link 软件的安装目录下。

| Choose Targe | et | | | ? × |
|--------------|------------|--------------------------------------|-----------|-----------------|
| Target Env | vironment: | 2 | | |
| Target | RDI | File | Version | Add |
| ADP | 1 | F:\ITDESI~1\ADSv1_2\Bin\Remote_A.dll | 1.2.0.805 | |
| ARMUL. | 1 | F:\ITDESI~1\ADSv1_2\Bin\ARMulate.dll | 1.2.0.805 | Remove |
| | | | | |
| | | | | Re <u>n</u> ame |



调试仿真

菜单>Execute

Go: 全速运行; Stop: 停止全速运行; Step in: 单步运行,跟踪到被调用函数里边去; Step: 单步运行,把被掉函数当成一整条简单的语句; Step out: 跟踪到函数里面后,可以退出到当前函数的调用处; Run To Cursor: 运行到光标处; Toggle Breakpoint: 对光标所在的行设置或清除断点。

| System Views | Execute Options Wind | ow <u>H</u> elp | | |
|--------------|---------------------------|-----------------|---------------------------------------|---------------------------|
| Ge 6e | <u>G</u> o Stop | F5 ShiffaE5 | I I 🖪 📻 🖂 🖻 | |
| | | SHILTIS | _ 1 | ;IMPORT main entry |
| v | Step In | F8 | 2 | AREA Init, CODE, READONLY |
| | Step | F10 | 3 | |
| 1. | Step Out | Shift+F8 | 4 | ENTRY |
| 1- | Run To Cursor | F7 | 5 | |
| 1. | | | - 6 | IMPORT main |
| ۰. | Show Execution Conte | xt | · · · · · · · · · · · · · · · · · · · | BL main |
| ۱. | Tagala Braakapint | FO | | END . |
| {• | roggie <u>p</u> reakpoint | ra Est | | |
| {. | loggle <u>W</u> atchpoint | Esta | | |
| {. | Set Watchpoint | | | |
| {. | Delete All Breakpoints | | | |

Part 2 用JLINK烧写FLASH

1、配置ADS工程
 2、编译生成.bin文件
 3、配置jflash工程
 4、烧写FLASH



配置ADS工程

菜单>Edit>DebugRel Settings

需要配置的选项:

1、Target Settings

- 2、ARM Assembler (编译器)
- 3、ARM C Compiler (编译器)
- 4、ARM Linker (链接器)

5、ARM fromELF

其他选项默认即可。



1、Target Settings

| DebugRel Settings | | | | ? × |
|---|--------------------------------------|------------------------|---|--------|
| Target Settings Panels | Target Settings | | | |
| □ Target ▲ □ Target Settings □ Access Paths □ Build Extras | Target Name: Linker: | DebugRel ARM Linker | × | |
| Runtime Settings | Pre-linker: | None | • | |
| File Mappings Source Trees ARM Target | Post-linker: -Output Directory: - | ARM fromELF | | |
| E- Language Settings | (Projecth | | | Choose |

目的:选择链接完成后,对文件进行操作。

链接完成后ADS会默认生成一个.axf的文件,为了得到.bin文件, 需要调用ARM fromELF命令将.axf转换为.bin文件。



4、 ARM Linker (**Output**)

| DebugRel Settings | | | ? × |
|--|------------------------------|---------------------------|--------------------------|
| Target Settings Panels | ARM Linker | | |
| ⊟ Language Settings ▲ ARM Assembler | Output Options L | ayout Listings Extras | |
| ARM C Compiler | Linktype C <u>P</u> artia | | pi 🗌 Reloc <u>a</u> tabl |
| - Thumb C Compiler | • Simple | | |
| □ Thumb C++ Com □ Linker | Scattered | Sp. | <u>11t Imag</u> |
| ARM Linker | Scatter | | Choose |
| Editor | Symbol | | Choose |
| Custom Keywords | Symbol <u>e</u> diting | | Choose |

目的:设置代码段的起始地址为FLASH的起始地址。



5、ARM fromELF



目的:转换为二进制文件并指定输出位置。

编译生成.bin文件

菜单>Projiect>Make

| | Check Synta <u>x</u> Preprocess Precompile | Ctrl+; | - () - M |
|-----------------|--|---------------|-----------|
| | <u>C</u> ompile | Ctrl+F7 | #define |
| Surger Starting | Disassemble | Ctrl+Shift+F7 | #define] |
| p05.mcp | Bring Up To <u>D</u> ate | Ctrl+U | void Dela |
| abugRal | Make | F7 | |
| cougher | Stop Build | Ctrl+Break | { void ex |
| 5 Link | Remove Object Code | Ctrl+- | IOPM |
| File | Re-search for Files | | { |





菜单>Options>Projiect settings

1、Target Interface

| Project settings | |
|--|--|
| General Target Interface CPU | Flash Production |
| JTAG | |
| JTAG speed before init | JTAG speed after init |
| <u>Auto selection</u> | <u>Auto selection</u> |
| ○ Adaptive <u>clocking</u> ○ 5 <u>k</u>Hz | C Adaptive <u>clocking</u> C 4000 Y <u>k</u> Hz |

这个根据需要选择,若不确定就选择自动。

配置jlink工程

菜单>Options>Projiect settings

| | Project settings | ×. |
|--|--|----|
| 2. CPU | General Target Interface CPU Flash Production | |
| | ○ Device ✓ Check core ID • Core ARM7 ID 00000000 | |
| 指定CPU开初始化 | Little endian I Use target <u>B</u> AM (faster) Addr 40000 4 KB 加快烧写返 | 度 |
| S3C4510 - 记事本 文件(E) 编辑(E) 格式(Q) 查看(V) 帮助(H) SETMEM 0x3FF0000, 0xE7FFFF90,32 SETMEM 0x3FF3010, 0x00003002,32 SETMEM 0x3FF3014, 0x02000060,32 SETMEM 0x3FF302C, 0x14010380,32 SETMEM 0x3FF303C, 0xCE338360,32 DAAGE AND | Use following init sequence: Use following init sequence: Use following init sequence: Add action Action Type Add action Action Type Address Data E7FFF90 Hex Comment OK Cancel | × |

配置jlink工程

菜单>Options>Projiect settings

3、FLASH

指定FLASH型号

| General 1 | Farget Interface | CPU Flash Pro | duction | | | > |
|---|---|--|---------|---|---|---|
| FlashBar | ik Bank[0] | * | Add | Be | emove | 3 |
| 🔲 Use cu | ustom <u>R</u> AMCode | | | | | |
| Base <u>A</u> de | dr 00000000 | <u>O</u> rganization 16 | ; 💌 Bit | s x 1 | ✓ Chip(s) | < |
| Autom | atically detect flash | | Sal | act flash devi | . 44 그나 | |
| (m) 75 | | モールロ | Jei | sor nasi i devi | ᅊᇏᅟᅳᅒ | / 1 |
| Manufacti | Coloct Bach dovi | | | ect nasiri <u>d</u> evi | ◎ - 年 フ | |
| Manufact e | Select flash devi | 第一 少 | | eet nasir gevi | ☞\$\$ | |
| Manufact o Ca | Select flash devi Manufacturer SS | ice T | | sot nasiri gevi | [™] ₩ | |
| Manufacto Ca | Select flash devi Manufacturer SS | ice T | | Size | ^{rce} | |
| Manufach Ca Start A | Select flash devi Manufacturer SS Manufacturer SST | Ce T Device SST39LF200A | | Size | wumSectors | |
| Manufach Ca S Start A | Select flash devi Manufacturer SS Manufacturer SST SST | Ce T Device SST39LF200A SST39LF400A | | Size 256 KB 512 KB | NumSectors 64 128 | BFC BFOU |
| Manufach C S Start A End A | Select flash devi Manufacturer SS Manufacturer SST SST SST | にCE IT Device SST39LF200A SST39LF400A SST39LF800A | | Size 256 KB 512 KB 1024 KB | NumSectors 64 128 256 | BFE BFOU |
| Manufach C S Start A End A | Select flash devi Manufacturer SS Manufacturer SST SST SST SST | にCE ICE IT Device SST39LF200A SST39LF400A SST39LF800A SST39LF160 | | Size 256 KB 512 KB 1024 KB 2048 KB | NumSectors 64 128 256 512 | BF2 BF0 BF0 BF0 |
| Manufacto C S Start A End A Selecteo | Select flash devi Manufacturer SS Manufacturer SST SST SST SST SST SST | にCE ICE IT Device SST39LF200A SST39LF400A SST39LF800A SST39LF160 SST39VF160 | | Size 256 KB 512 KB 1024 KB 2048 KB 2048 KB | NumSectors 64 128 256 512 512 | 8 FF |
| Manufacto C S Start A End A Selecteo | Select flash devi Manufacturer SS Manufacturer SST SST SST SST SST SST | Ce Device SST39LF200A SST39LF400A SST39LF800A SST39LF160 SST39VF1601 | | Size 256 KB 512 KB 1024 KB 2048 KB 2048 KB 2048 KB | NumSectors 64 128 256 512 512 512 | 8 FF BFD BFD BFD BFD BFD BFD BFD BFD BFD BFD |
| Manufacto C S Start A End A Selected | Select flash devi Manufacturer SS Manufacturer SST SST SST SST SST SST SST | Ce Device SST39LF200A SST39LF400A SST39LF800A SST39LF160 SST39VF1601 SST39VF1601 SST39VF1602 | | Size 256 KB 512 KB 1024 KB 2048 KB 2048 KB 2048 KB 2048 KB | NumSectors 64 128 256 512 512 512 512 512 | |
| Manufacte C Start A End A Selectec | Select flash devi Manufacturer SS Manufacturer SST SST SST SST SST SST SST SST SST | Ce ICE IT Device SST39LF200A SST39LF400A SST39LF400A SST39LF160 SST39VF1601 SST39VF1601 SST39VF1602 SST39VF200A 位在 | | Size 256 KB 512 KB 1024 KB 2048 KB 2048 KB 2048 KB 2048 KB 2048 KB 2048 KB | NumSectors 64 128 256 512 512 512 512 512 64 | |

烧写FLASH

1、连接实验板

| <u>File E</u> dit <u>V</u> iew | Target | <u>Options</u> | Window | H | elp | | | | Ę |
|--------------------------------|------------------------------|----------------|--------|----------------------|-------|----|----|----|---|
| | <u>Connect</u> Disconnect | | | PRANALA PLINAM | | | | | |
| RASH Project - ne | | | | D:\ADS-Projiects\sup | | | | up | |
| Name | Show CFI info | | | tress: 0x0 | | | | 1 | |
| Connection | | | | | - | - | | | 1 |
| Target interface | Test | | | + | ldre | ss | 0 | 1 | |
| | - | | 34230 | IVG | 3000 | Ø | 7E | 00 | ę |
| Init JTAG speed | Lock/Unlock sectors | | 00010 | | FE FF | | 1 | | |

J LOG

Connecting

- Connecting via USB to J-Link device O
- J-Link firmware: V1.20 (J-Link ARM V8 compiled Dec 1 2009 11:42:48)
- JTAG speed: 5 kHz (Fixed)
- Initializing CPU core (Init sequence) - Initialized successfully
- JTAG speed: 8000 kHz (Auto)
- J-Link found 1 JTAG device. Core ID: 0x1F0F0F0F (ARM7)
- Flash ID (Chip 0) OnBF234B (Verified)



2、加载要烧写的.bin文件

| J FIASH S | EGGEI | R J-Fli | ash ARN | I V4.02d | d - [new project *] |
|--------------|--------------|--------------|---------|-----------------|--------------------------------------|
| <u>F</u> ile | <u>E</u> dit | <u>V</u> iew | Target | <u>O</u> ptions | s <u>W</u> indow <u>H</u> elp |
| <u>0</u> | pen | | C | Ctrl+O | |
| Me | erge | | | | nash D:\ADS-Projiects\supervivi-128K |
| <u>S</u> e | ave | | ¢ | trl+S | |
| Sa | ave As. | 555 | | 1 | |





3、Program

| | | | | | ~~ | | 100 | |
|----------------|---------------------|----|------|-----|----|----|-----|----|
| CDU | Check <u>b</u> lank | F2 | 0040 | 00 | 20 | AØ | EÀ | |
| Endian | Fill with zero | | 0050 | 00 | 60 | AØ | EX | ŀ |
| Check core Id | Erase sectors | F3 | 0060 | 00 | AØ | AØ | E3 | Ì |
| Core Id | Frase chin | | 0070 | 80 | 19 | AЙ | E3 | |
| Use target RAM | Lines sup | | 0000 | 0.4 | 10 | | TO | F. |
| RAM address | Program | E5 | 0080 | 04 | 10 | 51 | Eq | 1 |
| RAM size | D 014 17 | | 0090 | 00 | 00 | 81 | E5 | > |
| | Program & Verify | F6 | ØØAØ | 00 | 30 | AØ | E3 | 5 |
| Flash memory | Auto | F7 | GORG | 60 | 70 | 98 | E5 | ſ |

| J-Flash A | RM V4.02d | | | × |
|-----------|---|--|--|-----------|
| ? | The programming You can erase of Do you want to e | area is not co roverwrite the trase the affec 否(<u>N</u>) | mpletely empty. affected area. ted area before 取消 | program ? |



祝大家学习愉快!

Thank You !